



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/818,214	03/26/2001	Rahul Sharma	SUNMP005	2476

25920 7590 06/24/2004

MARTINE & PENILLA, LLP  
710 LAKEWAY DRIVE  
SUITE 170  
SUNNYVALE, CA 94085

EXAMINER

LOHN, JOSHUA A

ART UNIT	PAPER NUMBER
----------	--------------

2114

DATE MAILED: 06/24/2004

8

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/818,214

Applicant(s)

SHARMA ET AL.

Examiner

Joshua A Lohn

Art Unit

2114

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☒ Responsive to communication(s) filed on 17 May 2004.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-32 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-20, 28, 29 and 32 is/are rejected.
- 7) ☒ Claim(s) 21-27, 30 and 31 is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 26 March 2001 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

## Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

## Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)  
Paper No(s)/Mail Date \_\_\_\_\_
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: \_\_\_\_\_

## **DETAILED ACTION**

### ***Response to Arguments***

Applicant's arguments filed 5/17/2004 have been fully considered but they are not persuasive.

In response to applicant's arguments with respect to claim 1 the examiner respectfully disagrees. It is true that Deshpande does not disclose the replicated state manager, however it would have been obvious to one skilled in the art of the invention that Deshpande would have had some form of replicated state manager. The existence and use of replicated states would obviously have required some form of management. The fail over operations disclosed on page 14 would have obviously required some form of state management.

Applicant's further argument that Deshpande does not suggest a replicated state manager that can use an in-memory database is also respectfully disagreed with. The disclosure of Deshpande does state that in-memory databases are not suitable for persistent components, but the previous statement is that the in-memory database is generally used in the replication of session states. This general use, as disclosed on page 10, shows the known common use of in-memory storage for replication of session state information as required in claim 1 of applicant's application. Deshpande's statement that an in-memory database may not be suitable does not teach away from the use of an in-memory database because he states that it is still generally used despite this flaw.

In response to applicant's argument that Deshpande does not discuss usage of both an in-memory database and a replicated state server, which can both contain entity bean states that can be recovered, the examiner respectfully disagrees. The teaching of a Servlet session, on page 10,

Art Unit: 2114

provides a system that stores replicas of the system state across multiple nodes, and this information is generally stored in an in-memory database. These multiple nodes act as replicated state servers, all of which can use the in-memory databases to recovery entity bean states, which are part of the system state.

In response to applicant's arguments with regard to claim 11, the examiner respectfully disagrees. With respect to the arguments involving the in-memory database, the examiner has previously explained that Deshpande discloses that, despite any flaws, the in-memory database is still generally used for state replication. With respect to the arguments of having in-memory storage in addition to the replicated state server, the examiner explained previously that the Servlet session, taught by Deshpande, discloses an in-memory storage replicated across numerous servers.

In response to applicant's arguments with regard to claim 16, the examiner respectfully disagrees. With respect to applicant's argument that Deshpande does not suggest storing a state of an entity bean within a state object of a managed state part, the examiner feels that state object is the containers, which include the replicated entity beans. These containers would obviously include the state of an entity bean to allow for the replication and fail over. All the containers are included in a managed state part, as they are managed by the same session, this can all be seen in the example of section 8.4 on page 15.

With respect to applicant's argument that the use of two entity beans in the same application session does not constitute a relationship, the examiner respectfully disagrees. The

Art Unit: 2114

phrasing of claim 16 does not provide any explicit definition of the relationship that is shared between the two entity beans. The use of two entity beans in the same application session is the relationship of sharing the same session. This relationship is sufficient to meet the limitations as currently claimed.

With respect to applicants argument that Deshpande only teaches of having one copy of a past state of the entity bean, and fails to teach of a state object that stores a state of an entity bean and also includes the feature of a replicated state server that stores a replica of the state server, the examiner respectfully disagrees. Deshpande discloses the storage of the entity bean state in two locations, the state object storing the state of an entity bean in the object currently executing the bean, and the state object of a replicated state server that stores the state is the shared database for use in failing over an entity bean, see section 8.2 of page 14.

In view of the above response the examiner maintains the rejections as previously stated, and reiterated below. Claims 21-27, 30, and 31 would still be allowable if rewritten in an independent form.

### ***Claim Rejections - 35 USC § 103***

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

Art Unit: 2114

Claims 1-20, 28, 29, and 32 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Clustering: Transparent Replication, Load Balancing, and Failover" by Salil Deshpande, published January 2000.

As per claim 1, Deshpande discloses executing a Java application on a server, where the Java application includes multiple entity beans, see Figure 1 on page 5, where plural entity beans (EJBs) are executed on the application server. It is obvious that Deshpande discloses a replicated state manager, wherein the replicated state manager would obviously have included includes program instructions for managing a replicated and migration capable session state of the Java application using an in-memory database within a Java server process.

This is obvious in the fact that Deshpande provides for replicas of session beans and the ability to transparently fail over to these replicated beans, and any replication and fail over would obviously be controlled by some form of a replicated state manager, see page 14. Deshpande additionally provides for the session state of an entity bean being stored within a database, see page 14, which it would be obvious to provide in-memory, see page 10.

It would be obvious to use an in-memory database for replication when using a system that has Servlet session state replication for fail over. Deshpande also obviously provides for the replicated state manager including replicating the in-memory state of the Java application to the replicated state server, see page 10, where when using a Servlet session the state is replicated across multiple nodes, this state replication obviously including any entity beans that are part of the session state.

As per claim 2, Deshpande discloses replicating state server data in-memory, see page 10.

As per claim 3, Deshpande teaches that it would be obvious to have a disk-based replication. This would have been obvious because an in-memory replication is more susceptible to failures and not suitable to persistent components.

As per claim 4, Deshpande discloses state replication to different types of state servers, see page 15, where a failure of the currently used state server causes the use of a distributed state server environment that holds the duplicated states.

As per claim 5, it is obvious that the state recovery occurs during application failure, this is obvious because on page 15 Deshpande discloses the fail over occurring during a session that is invoking various entity beans. Any session that is currently accessing entity beans is obviously running an application that is utilizes those entity beans.

As per claim 6, Deshpande discloses a checkpoint policy used to configure the fail over of the entity beans. The checkpoint policy is described as having a correct state provided at the end of each successful transaction, and storing this state in the database, see page 14.

As per claim 7, Deshpande discloses the storing of a state of the entity bean using a state object, where the state object is the correct state stored in the database, see page 14.

As per claim 8, It is described in the rejection of claim 1 above that the states are taught as being able to be stored as in-memory objects, it is obvious that the state is managed with a J2EE server process. This is obvious because the use of entity beans, as taught by Deshpande, requires that the J2EE standards be used, see page 4.

As per claim 9, Deshpande teaches of a logical separation between the application and the state objects, see page 15, where the application is accessing entity beans in one container while the state objects that include fail over duplicates exist in a separate logical container.

As per claim 10, Deshpande teaches updating and tracking changes of the state objects, for the checkpoint mechanism, in response to changes in the state of the application. This is shown in the state objects being checkpointed to the state of the last successful transaction, implemented by the application, see page 14.

As per claim 11, Deshpande discloses an application part having an entity bean object, see Figure 1. Deshpande obviously teaches a managed state having an object that stores a state of the entity bean object within a memory address space of a J2EE server process. This is obvious because Deshpande discloses the use of entity beans, as shown above, and this use of entity beans would require the use of a J2EE process on the server which is executing the objects, see page 4. Deshpande also discloses a replicated state server that stores a replica of the state object, where the replicated server is the database that stores the correct state of an entity bean, see page 14.

As per claim 12, Deshpande discloses a memory replicated state server, in the form of in-memory replication that would be obviously used if a Servlet session is being used with fail over, see page 10.

As per claim 13, Deshpande teaches that it would be obvious to have a disk-based replication. This would have been obvious because an in-memory replication is more susceptible to failures and not suitable to persistent components.

As per claim 14, Deshpande teaches of a logical separation between the application and the managed state objects, see page 15, where the application is accessing entity beans in one container while the state objects that include fail over duplicates exist in a separate logical container.



As per claim 15, Deshpande teaches of the replicated state manager tracking updates to the state object in response to changes in the state of the entity bean object as part of a transaction. This is shown in the replicated state being a state of the last successful transaction, which would obviously mean that the replicated state is updated as part of a transaction completing, see page 14.

As per claim 16, Deshpande discloses an application part having a plural entity bean objects, see figure 1. These entity beans are taught to be related, see the example on page 15, where entity beans E1 and E2 are related due to use by the same application session. Deshpande also teaches of a managed state having state objects to store the state of each of the entity beans, wherein the relationship of the beans is stored. This is shown in the example on page 15 with the state of the two entity beans being used by the invoking session, where an invoking session would obviously use the state of the entity bean objects to allow for proper replication and failover. Deshpande also teaches of a replicated state server that stores a replica of the first state object and a replica of the second state object, see the database on page 14 that stores a replica of the state of each entity bean.

As per claim 17, Deshpande discloses a memory replicated state server, in the form of in-memory replication that would be obviously used if a Servlet session is being used with fail over, see page 10.

As per claim 18, Deshpande teaches that it would be obvious to have a disk-based replication. This would have been obvious because an in-memory replication is more susceptible to failures and not suitable to persistent components.

Art Unit: 2114

As per claim 19, Deshpande teaches of a logical separation between the application and the managed state objects, see page 15, where the application is accessing entity beans in one container while the state objects that include fail over duplicates exist in a separate logical container.

As per claim 20, Deshpande teaches of the replicated state manager tracking updates to the state object in response to changes in the state of the entity bean object as part of a transaction. This is shown in the replicated state being a state of the last successful transaction, which would obviously mean that the replicated state is updated as part of a transaction completing, see page 14.

As per claim 28, Deshpande discloses a checkpoint policy used to configure the fail over of the entity beans. The checkpoint policy is described as being configured to have a correct state provided at the end of each successful transaction, and storing this state in the database, see page 14.

As per claim 29, Deshpande discloses a synchronous checkpoint method in which the checkpoints are stored as they are generated in response to the successful completion of a transaction, see page 14.

As per claim 32, Deshpande discloses the checkpoints being made in response to single transactions, see page 14.

*Conclusion*

**THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire **THREE MONTHS** from the mailing date of this action. In the event a first reply is filed within **TWO MONTHS** of the mailing date of this final action and the advisory action is not mailed until after the end of the **THREE-MONTH** shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than **SIX MONTHS** from the mailing date of this final action.


Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua A Lohn whose telephone number is (703) 305-3188. The examiner can normally be reached on M-F 8-4.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Robert Beausoleil can be reached on (703) 305-9713. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2114

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JAL



**SCOTT BADERMAN**  
**PRIMARY EXAMINER**